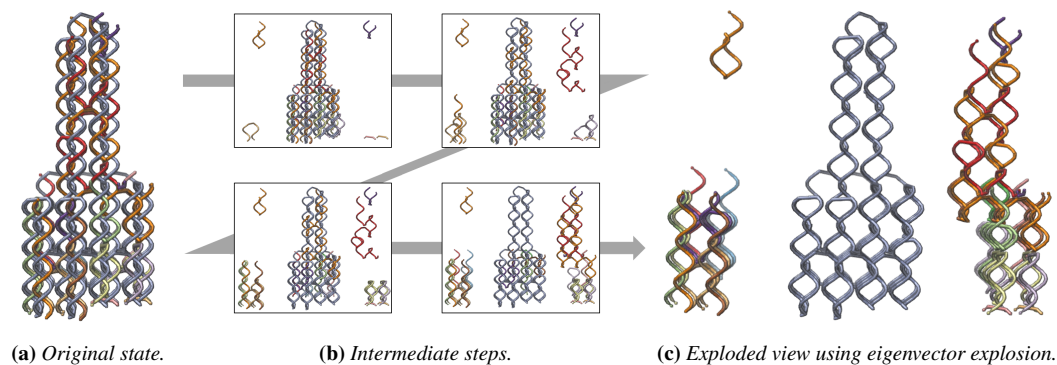


# Interactive Exploded Views for Molecular Structures

M. Sbardellati<sup>1</sup>, H. Miao<sup>1,2</sup>, H.-Y. Wu<sup>1</sup>, M. E. Gröller<sup>1,3</sup>, I. Barisic<sup>2</sup>, and I. Viola<sup>1,4</sup>

<sup>1</sup> TU Wien, Austria <sup>2</sup> Austrian Institute of Technology, Austria

<sup>3</sup> VRVis Research Center, Vienna, Austria <sup>4</sup> King Abdullah University of Science and Technology (KAUST), Saudi Arabia



**Figure 1:** A comparison of (a) the original, (b) the intermediate steps, and (c) the exploded view of a DNA origami tube, namely NanoTube, generated using the proposed sequential eigenvector explosion approach. This NanoTube is composed of one scaffold strand (grey) and several staple strands (other colors), whose nest structures can be clearly investigated through our techniques.

## Abstract

We propose an approach to interactively create exploded views of molecular structures with the goal to help domain experts in their design process and provide them with a meaningful visual representation of component relationships. Exploded views are excellently suited to manage visual occlusion of structure components, which is one of the main challenges when visualizing complex 3D data. In this paper, we discuss four key parameters of an exploded view: explosion distance, direction, order, and the selection of explosion components. We propose two strategies, namely the structure-derived exploded view and the interactive free-form exploded view, for computing these four parameters systematically. The first strategy allows scientists to automatically create exploded views by computing the parameters from the given object structures. The second strategy further supports them to design and customize detailed explosion paths through user interaction. Our approach features the possibility to animate exploded views, to incorporate ease functions into these animations and to display the explosion path of components via arrows. Finally, we demonstrate three use cases with various challenges that we investigated in collaboration with a domain scientist. Our approach, therefore, provides interesting new ways of investigating and presenting the design layout and composition of complex molecular structures.

## CCS Concepts

• **Human-centered computing** → **Scientific visualization**; Visualization toolkits;

## 1. Introduction

Molecules, in particular DNA, are the basis of all life on our planet. Although molecular biology has been long investigated, modern technologies further allow researchers to explore this field more effectively and deeply than ever before. One typical field that enjoyed a massive upturn thanks to the increasing computing and visualization capabilities is DNA nanotechnology. DNA nanotechnology

deals with the creation of DNA nano-structures which do not use DNA to carry genetic information but for the construction of arbitrary nanoscopic structures [ZNL14]. The resulting objects can, for example, be used to transport medication to specific cells in the body [DBC12].

According to Kozlíková et al. [KKF<sup>+</sup>17], visual clutter and occlusion are two of the main problems that need improvement when

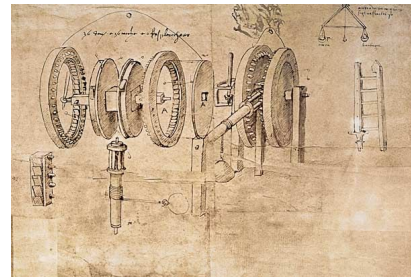
visualizing large molecular structures. Larger molecular structures, like DNA nano-structures, can consist of several hundred components, which are intertwined given the double-helical form of DNA [WC<sup>+</sup>53]. The resulting visual clutter and occlusion make it difficult for domain experts to get an in-depth understanding of these structures and the spatial relationships between their components from a given visualization. Additionally, occlusion lead to a problem in the *design process* of DNA nano-structures. Often domain experts cannot isolate the component that is being worked on without losing information about the surrounding objects.

Out of a large number of available approaches for handling above problems [ET08], we suggest the *exploded views* technique is one of the best suited for visualizing molecular structures and especially DNA nano-structures, due to the following aspects: Exploded views divide an object into several components and then displace them to make components that were occluded previously visible as shown in Figure 1. Exploded views reduce occlusion and clutter while still showing all components. This allows domain experts to keep an overview of a DNA nano-structure while being able to more closely investigate single components, helping them in the design and analysis of these structures. Exploded views do not deform components, which is an undesirable effect if the user wants to examine the component's shape. Furthermore, they generate additional information on how different components are connected. This helps domain experts in the exploration and presentation of the spatial structure and hierarchical component relationships of DNA nano-structures.

Other approaches, like transparency [ET07] or X-ray tunnels [BH04], can be used to solve the occlusion problem. We want to provide the user with additional information into component relationships. As the conventional approaches do not provide this, they are not suitable in our case. Other techniques like adaptive cutaways [BF08] or deformations [MTB03] are also not specifically designed for visualizing molecular structures. This is because they deform the structures and do not show all the components simultaneously, and thus cannot provide sufficient information about component relationships.

We present an interactive framework for generating exploded views that helps domain experts in the design and inspection of molecular structures. Furthermore, it provides experts with a visualization tool-set for presenting molecular structures to others. Our work can be divided into two separate approaches for creating exploded views, the first one being the *structure-derived exploded view*. This approach allows scientists to design of exploded views using the structure of the given object for calculating the key parameters efficiently. For creating more sophisticated exploded views, we present the *free-form exploded view*, where the user can manipulate the key parameters arbitrarily. As a tool for presenting the generated exploded views, we incorporated animations in our work. Our approach is implemented within SAMSON [ND16], which is an established tool for modeling and visualizing molecular structures. Our scientific contributions can be summarized as:

- an automated pipeline for creating static and animated exploded views of complex molecular structures to help scientists in the exploration and presentation of these structures,



**Figure 2:** Exploded view of a gear assembly from Leonardo da Vinci's Codex Atlanticus (15th century).

- an interactive method for creating multi-step custom explosion paths to depict complex explosion scenarios,
- an exploded-view based visualization concept that depicts relationships of different components within molecular structures,
- an enhancement of the design process of DNA nano-structures by enabling domain experts to isolate parts of their design.

### 1.1. DNA nanotechnology

DNA nanotechnology is a field that has been around for over 35 years. Seemann et al. [SK83] have been the first ones to describe an approach of using structural properties of DNA for constructing nanoscopic structures. The late 2000s have then seen a major increase in scientific activity. In 2006 Rothemund developed the *DNA origami* technique [Rot06], which revolutionized DNA nanotechnology and raised interest in the field [ZNL14]. DNA nanotechnology is appreciated for its potential application in medicine and biotechnology. Besides, computer-aided design and simulation tools gave domain experts the possibilities for fast prototyping and interactive visualization. DNA origami structures consist of a long single strand (scaffold) and several short single strands (staples). Using Watson-Crick pairings [WC<sup>+</sup>53], the single-stranded DNA are bound together. By specifying sequences of staples and their binding regions on the scaffold, strands will self-assemble into the target shape. The most prominent software for designing such structures is caDNAno [NC16].

### 1.2. Exploded Views

Exploded views are old techniques for handling visual occlusion. They were used by Leonardo da Vinci in the 15th century (see Figure 2). The most common use cases for exploded views over the last five centuries were construction manuals, architectural plans, and drawings in the field of mechanics, biology, and medicine. In these fields, it is advantageous to visualize the relationship between object components and still have an overview on the whole object. This feature is also one that makes exploded views well suited for visualizing molecular structures and DNA nano-structures.

Using computer tomography or computer-aided design of mechanical parts in combination with the visualization capabilities of current hardware for data acquisition, exploded views do not need to be drawn manually anymore. In some cases, they do not even need to be designed interactively on a computer, but can be generated fully automatically. In the late 1980s, the first

approaches in this respect were presented by various research groups [DFW87, K LW89, MK93, RKSZ94]. In Section 2 several relevant approaches will be discussed.

## 2. Related Work

**Explosion techniques:** To ensure the correct interpretation of exploded view diagrams by the users, it is important to follow certain design principles, as stated by Agrawala et al. [APH<sup>+</sup>03]. First of all, one needs to differentiate between structural diagrams and action diagrams. *Structural diagrams* show the components of an object in their final exploded position with no additional information. *Action diagrams*, on the other hand, add information about the operation, that was necessary to move the components to their final position. Action diagrams are sometimes considered to be superior to structural diagrams. We adopted the idea of action diagrams by adding *arrows* that show the path from the components' original position. Additionally, the *component hierarchy* plays an important role in the human perception of objects. This leads to another design principle, where multiple *step-by-step exploded views* are preferable over a *single view*. Step-by-step exploded views illustrate the process of the explosion, while single views show all the components already at their exploded position. We adopted these principles using the hierarchy to select explosion components and to define the order in which the components should be exploded.

Creating exploded views in a digital environment creates possibilities to enhance them using interaction methods. Li et al. [LACS08, LAS04] introduced *animated exploded views* and interaction methods like *direct manipulation*, *riffling*, or the *automatic exposure* of target parts. Their use of *blocking constraints* is obstructive when exploding intertwined molecular structures. Our approach provides animated exploded views as well as the interactive movement of components via a slider. We create own explosion paths for selected components and do not use blocking constraints.

In the field of molecular visualization, the explosion components are often composed of multiple sub-objects. Therefore, it is necessary to define a *representative point* for the component to use in position calculations. Sonnet et al. [SCS04] present three different approaches to this problem. One can use the center of the objects bounding box or a significant point of its skeleton, if available. The third option is to combine the two approaches and use a point on the skeleton, which is closest to the bounding box center. For molecular data, it does not make sense to compute skeletons. Therefore, we use the average 3D position of all the component's atoms, to calculate a representative point.

The calculation of the explosion direction and distance are two of the main aspects when computing exploded views and can be done in several ways. Elmqvist's [Elm05] *BalloonProbe* defines a sphere with a user-defined radius. All the objects inside the sphere are displaced to the rim of the sphere along the vector that passes from the sphere center through the component center. Inspired by this, we develop a *spherical exploded view* in our approach. Tatzgern et al. [TKS10] compute 3D directions in which the component would not be blocked by other objects to get the explosion direction. Since smaller components do not need to be moved as far as larger ones, they compute the *distance* for each component individually. We

adopted this idea for calculating the explosion distance so smaller objects move smaller distances as compared to than larger ones. Sometimes it is also advantageous to *split components* to create a better view [BG06, LACS08]. Exploding along the eigenvectors, components are moved in opposite directions if they are on different sides of the object center.

**Visual complexity reduction in molecular data:** The visualization of molecular data is a highly complex problem. Exploded views are one of the few techniques that are not yet widely used to address this issue [KKF<sup>+</sup>17]. They can be used to a great effect as shown by *Metamorphers*, an approach by Sorger et al. [SMR<sup>+</sup>17]. It is an animation framework, which allows the user to create reusable and flexible animations to morph a molecular object into desired representations, one of them being exploded views. Their results are visually appealing and show the inner structures and hierarchies of the exploded object in an expressive way. In our approach, we adopt animations in a similar way.

Other approaches creating visual representations of molecular data are more applied [KKF<sup>+</sup>17]. Methods such as contour lines [LVRH07, KBE09], ambient occlusion [TCM06], and depth darkening of distant objects [LCD06] are often used. Computing the best viewpoint for a given object also enhances the visual representation [VFSL02]. Nadezhda et al. [DKDA11] reduce the visual complexity of molecular data by generating network graphs, which can be compared to spot differences between similar molecules.

**Visualization of DNA nano-structures:** A visual representation specialized in DNA nano-structures was developed by Miao et al. [MDLS<sup>+</sup>18]. Their approach features a multiscale visualization, which emphasizes on showing different semantic levels of abstraction. A continuous transition between ten possible scales, from an atomic level up to showing the geometry of the whole object, is suggested. The reduced complexity of the DNA nano-structure facilitates its understanding and helps domain experts during the design process. We incorporate their visual representation in the presentation of our results to enhance comprehensibility.

## 3. Exploded View Visualization

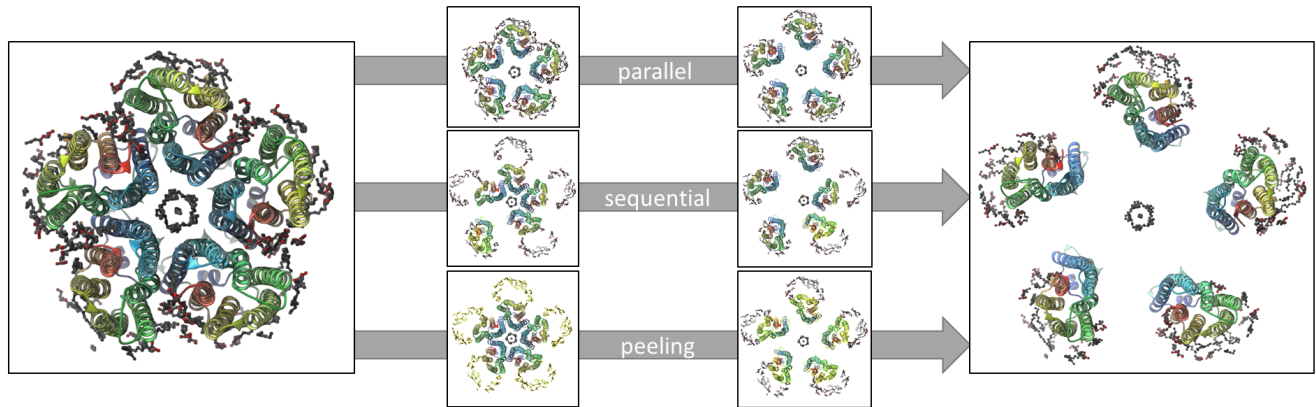
Out of their essence, we derived four key parameters for creating exploded views. First, elements of an object are specified that need to be exploded. Therefore, a *selection process of explosion components* has to be defined. When animating exploded views, the *order* in which components explode is of interest. Additionally, it needs to be known how far and in which direction the components should be moved. An *explosion distance*, in pico-meter, and *direction* is computed for this purpose. In the following sections, we will explain our two approaches, i.e. the *structure-derived exploded view* and the *free-form exploded view*. We describe in detail how the four mentioned parameters are derived for each of the approaches.

### 3.1. Structure-Derived Exploded View

The goal of the structure-derived exploded view is to generate an exploded visual representation based on the structural object information. All parameters for the exploded view are derived from the object's hierarchical structure and the position of the explosion

	selection of components	explosion order	explosion distance	explosion direction
Structure-Derived Exploded View	according to structure hierarchy	parallel, peeling, sequential	calculated using the spatial information of the structure	spherical or along an eigenvector
Free-form Exploded View	arbitrary explosion groups	parallel	defined by user drawn path	defined by user drawn path

**Table 1:** Overview of the four key parameters for the two exploded view approaches.



**Figure 3:** The three explosion orders. The parallel explosion moves all components at once, the sequential explosion one component after another, and the peeling explosion moves the components in layers.

components (see Table 1). The two sub-styles concerning explosion direction are the *spherical explosion* and the *eigenvector explosion*. The two sub-styles differ only by their explosion direction. The other parameters are computed and used equally.

### 3.1.1. Selection of Components

Data containing molecular structures are usually already assembled in a hierarchical way. To handle the selection of explosion components, a parent-child relationship is used. The user can select any hierarchical level as parent and child level as long as the child level is at least one level below the parent level in the hierarchy. We consider each object at the parent level as an individual explosion object and the children as its explosion components. To calculate the three other key parameters for the children, only the structure of their parent node is taken into account.

In the following, we refer to explosion objects as parents and explosion components as children. We also need to define a representative point for the parents and children if positional information is required in the calculations. We use the *center* of the parent or child structure, which we define as the average 3D position of all comprising atoms.

### 3.1.2. Explosion Order

Depending on the user goal, exploding all the children simultaneously is not enough. Therefore, we present three different explosion orders: parallel, sequential, and peeling.

The *parallel explosion* is a fast way to create an overview on an object's structure by exploding all the children simultaneously (Figure 3 first row). This allows the user to quickly get an understanding of how many components there are in the object and where

they connect in the unexploded state. On the other hand, this approach makes it hard to focus on single components since they all move simultaneously.

To improve this, we present *sequential explosion*. We move one child after another, starting with the one that is the farthest away from the parent center. The children are sorted by their distance to the parent center before they are exploded one by one (Figure 3 second row). This approach, especially animated, highlights the spatial relationship of the currently moving component to its neighbors and provides a good overview of how each part fits into the whole object.

The objective underlying the *peeling explosion* is to move the children in layers. The number of layers  $n_l$  is chosen by the user to be between one and ten. To assign a layer for each child, we first need to calculate the length of peeling intervals  $l_p$ . This is done by subtracting the distance of the nearest child to the parent center  $d_j$  from the distance of the farthest child  $d_k$  and then dividing the result by  $n_l$ :  $l_p = \frac{d_k - d_j}{n_l}$ . With the peeling intervals and the number of layers, we now can assign children to the individual layers. After this mapping is done, we can now explode one layer after the other (Figure 3 third row). This approach is preferable if one wants to inspect occluded parts without exploding all components.

### 3.1.3. Explosion Distance

The *explosion distance*  $d_i$  defines how far each child  $i$  is moved. It is calculated separately for each  $i$ , to include the children's position in the distance calculation. It consists of the following four distance parameters:  $d_i = e \cdot f \cdot s \cdot p_i$ .

- The main source of changing the explosion distance interactively is a slider, which is controlled by the user and linearly manipulates the *explosion value*  $e \in [0, 99]$ .

- Another slider controls the *force parameter*  $f \in [0.2, 3]$ , which adjusts the maximum explosion distance.
- To ensure that  $d_i$  is less for smaller objects than for larger ones, we introduce the *structural parameter*  $s$ .  $s$  is calculated by dividing the average distance of all the parent's atoms to its center by a given constant  $sD$ . The value 25 for  $sD$  was empirically determined to give the best results. The three already mentioned parameters are the same for each child of a parent.
- The *position parameter*  $p_i \in [0, 1]$ , ensures that children that are closer to the parent center are not exploded as far as children that are farther away and differs for each child.  $p_i$  is computed by normalizing the distance of child  $i$  to its parent in the unexploded state  $do_i$  in the interval between the farthest,  $\max(do_i)$ , and the nearest,  $\min(do_i)$ , child to the parent:  $p_i = \frac{do_i - \min(do_i)}{\max(do_i) - \min(do_i)}$ . Using this parameter has the additional effect that the underlying shape of the object is not distorted, since all children explode relative to their original position. It is also possible to omit the position parameter and just use the other three parameters. For linear or tube-like objects,  $p_i$  has the effect that children near the object center do not move far enough.

### 3.1.4. Spherical Explosion

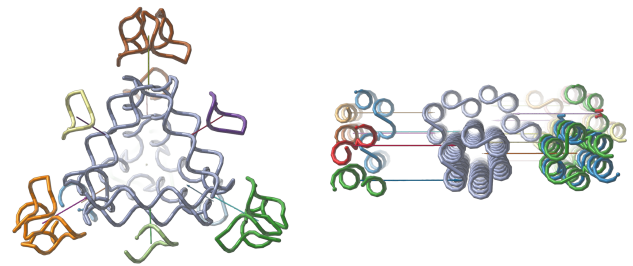
The *explosion direction* for the spherical explosion is defined by the normalized vector  $\vec{v}_i$  from the parent center  $c_p$  to the child center  $c_i$ . This approach is inspired by the balloon probe of Elmqvist [Elm05]. Instead of projecting the children to a sphere, we move them along  $\vec{v}_i$  according to the previously computed explosion distance  $d_i$ . If the user moves the explosion slider, the updated position of a child  $\bar{c}_i$  is calculated by multiplying  $\vec{v}_i$  with the updated distance of the child to the parent center  $\bar{d}_i = do_i + d_i$  and then moving  $\vec{v}_i$  to start at  $p$ :  $\bar{c}_i = \vec{v}_i \cdot \bar{d}_i + c_p$ .

Finally, we move all the atoms of child  $i$  to their new position. This is done by calculating the vector from  $c_i$  to  $\bar{c}_i$  and adding it to the position of every atom of  $i$ . Moving the children in a spherical way, this type of exploded view works best with objects that have similar extents in all three dimensions, as shown in Figure 4a.

### 3.1.5. Eigenvector Explosion

Eigenvector explosion uses eigenvectors to determine possible *explosion directions*. To calculate a parents eigenvectors, we first need to build a covariance matrix from our dataset. In our case, a parent consists of  $n$  children, each having coordinates in three dimensions, giving us  $j = 3$  features for each child. Since the number of eigenvectors of a matrix is limited by  $\min(n - 1, j)$ , we can compute a maximum of three possible explosion directions. The eigenvectors can be used for this purpose because they tell us the directions in which the data-set has the most variance and therefore in which direction it makes sense to move the components. The first eigenvector is the one with the largest variance (highest eigenvalue). The following eigenvectors have the largest possible variance under the constraint of being orthogonal to the previous ones.

In a preprocessing step, we build the covariance matrix from our data-set, i.e., the  $\vec{v}_i$ s. We then compute the three eigenvectors  $\vec{e}_x, x \in \{1, 2, 3\}$  and normalize them. The user can then decide along which of these directions they want to move the children. The movement is done by extending  $e_x$  by the explosion distance  $d_i$ . Additionally,



(a) Spherical exploded view of a DNA tetrahedron. (b) Eigenvector exploded view of a DNA origami tube along the second eigenvector.

**Figure 4:** Structure-derived exploded views. The arrows indicate the explosion distance and direction. We use the DNA visualization of Miao et al. [MDLS<sup>+</sup>18].

a plane is positioned orthogonally to  $\vec{e}_x$  through the parent center. For children on the front side of the plane, the extended vector is added to and for children on the backside, it is subtracted from the original child position  $c_i$ :  $\bar{c}_i = c_i \pm \vec{e}_x \cdot d_i$ .

To move all the child atoms to their new position, we calculate the vector from the old child center to  $\bar{c}_i$  and add it to all the atoms of the child. As seen in Figure 1c and 4b, the eigenvector explosion can be used and provides pleasing results with objects of all shapes.

## 3.2. Free-form Explosion

For certain use cases, like simulating the docking process of a ligand to a protein, it is necessary to specify custom explosion paths. Therefore, we introduce *free-form explosion*. The user can manually select the explosion components as well as the explosion direction and distance and is not bound by the underlying object structure (see Table 1). Using this approach, it is possible to create more complex exploded views and animations.

### 3.2.1. Selection of Components

Instead of using the atom groups provided by the given hierarchy of the molecule, users can select arbitrary collections of atoms and mark them as an *explosion group* following the coming constraints: First, a new explosion group is not allowed to contain atoms that are already part of another explosion group. It is not possible for an atom to explode in two directions at the same time. Additionally, an explosion group is not empty. The same as in the structure-derived explosion, we need a representative point for the group i.e., its center, which again is the average of the atom positions.

### 3.2.2. Explosion Order

Since it is possible to pack an arbitrary collection of atoms into a group, it often does not make sense to sort the groups according to their distance to the object center, since their representative points do not give useful information about the location of the group. For example, if one puts two atoms into a group that lies on the outside of the object but on opposite sides, the group center would be very close to the object center. When sorting the groups and exploding

them, this group would probably explode very late in the process, even though its single atoms are on the outside of the object. Therefore, in this approach only the *parallel explosion* order is used i.e., all groups explode simultaneously.

### 3.2.3. Explosion Distance and Direction

For the free-form explosion, the *explosion distance* and *direction* are interactively defined. This is done directly by the user by drawing an explosion path for each group into the viewport. Each path is a sequence of arrows with the first one starting at the center of its explosion group and ending wherever the user clicks in the viewport. The successive arrows always start at the end of the previous ones. The main source of interaction is an explosion slider that triggers the movement of the groups along their respective explosion paths. The explosion distance of each group is defined by its path.

To allow the user to *manipulate a path* after it was initially drawn, a start marker  $m_{i,1}$  at the arrows starting-point and an end marker  $m_{i,2}$  at its end-point is generated for each arrow, as illustrated in Figure 5a. By changing the position of these control markers, the path can be adjusted. The arrows for an explosion group are saved in a list, sorted from the first to the last arrow. Each arrow  $i$  is a tuple  $a_i = (m_{i,1}, m_{i,2}, l_i)$  with  $m_{i,1}$  and  $m_{i,2}$  being the arrow's control markers and  $l_i$  being the length of the arrow. It is also possible to delete the last arrow from the list, to reduce the explosion path.

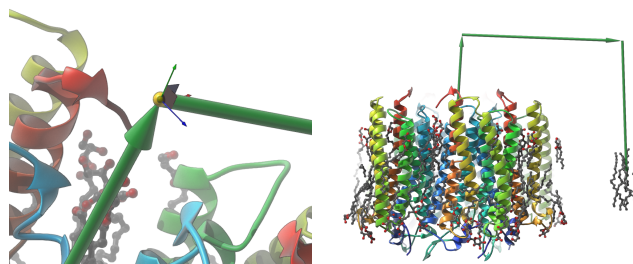
Moving an explosion group requires the calculation of its position  $pg$  on the corresponding explosion path. First, the relative position of the group along the explosion path  $\pi_g \in [0, 1]$  is determined by the position of the explosion slider. Next, we calculate the total length  $l_t$  of the path which is the sum of the length  $l_j$  of all its arrows  $a_j$ :  $l_t = \sum_{i=1}^n l_i$ . To know at which length  $l$  of the path the group needs to be moved, we multiply the total length with the percentage the slider was moved  $l = l_t \cdot \pi_g, \in [0, l_t]$ . In the next step, we compute on which of the arrows the group will land. This is done by determining the arrow  $a_j$  so that:  $\sum_{i=1}^{j-1} l_i < l \leq \sum_{i=1}^j l_i$ . To get the group's position along the explosion path corresponding to  $l$ , we calculate the relative position  $\pi_j \in [0, 1]$  of the group on arrow  $a_j$  by subtracting the length of all arrows before  $a_j$  from  $l$  and dividing the result by  $l_j$ :  $\pi_j = \frac{l - \sum_{i=1}^{j-1} l_i}{l_j}, \in [0, 1]$ .

Next, we determine the vector  $\vec{v} = m_{j,2} - m_{j,1}$  and resize it to the needed length by multiplying it with  $\pi_j$ . By adding  $m_{j,1}$  to  $\vec{v}$ , we translate the vector back to the arrow's start position. This creates a vector that points to  $pg$ . To move all the atoms of the group to their new position, we subtract the position of its center from  $\vec{v}$  and then add the result to the position of every atom of the group. An example of a free-form exploded view is shown in Figure 5b.

## 4. Interaction

To give the user additional control over the design and animation of the exploded view, we provide a set of interaction methods:

- The main mechanism for interactively controlling an explosion is the *explosion slider*. By manipulating this slider, the explosion components are moved. If a group of explosion components is selected, only these components will explode, providing the opportunity of moving only one or a specific set of components.



(a) The manipulation of the arrows is possible by moving the marker. (b) Exploded view that extracts the ligand from the center of the pump.

**Figure 5:** Exploded views of a sodium pump KR2 protein (PDB 6RF3) [KPG<sup>+</sup>19] using free-form explosion. The secondary structure visualization is provided by SAMSON [ND16].

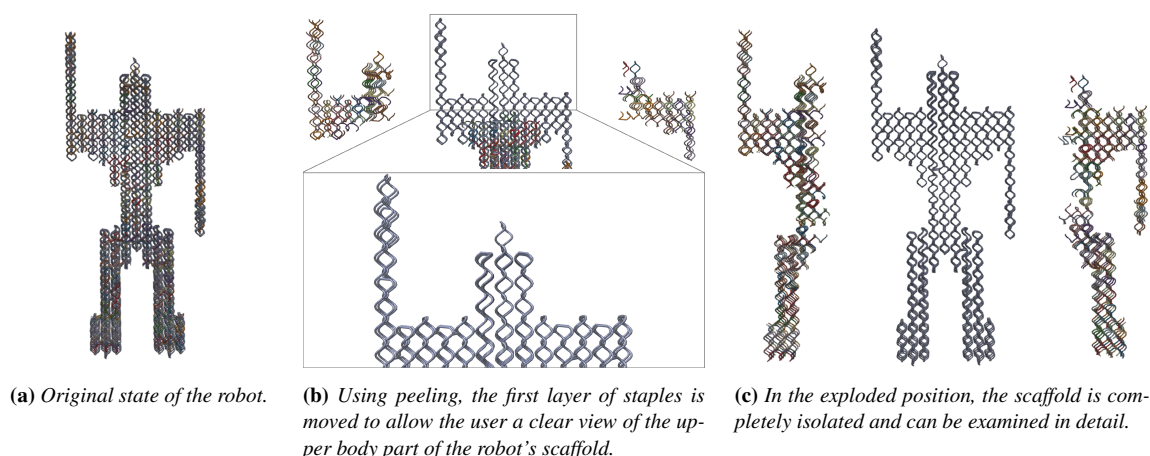
- It is also possible to toggle the visibility of *arrows* that indicate the explosion paths. When using free-form exploded views, they are shown by default and can also be manipulated. The structure-derived exploded view does not show the paths by default and if they are shown they cannot be changed since the path is fixed.
- *Animated exploded views* are a tool for presenting the composition of molecular structures or DNA nano-structures. For controlling the animation, we provide + and - buttons, which move the explosion slider automatically to the right (+) or left (-). The slider is moved continuously for a user-defined *step length* at a user-defined *time interval*. With these features, a user can control how fast and continuously the components will move.
- To further enhance the animations, we provide a set of *ease functions*. They allow the user to move the components in a more realistic way. Instead of just exploding at a *linear* speed, the user can choose between the following acceleration modi: *cubic ease-in* (first slow, then fast), *cubic ease-out* (first fast, then slow), and *cubic ease-in-out* (first slow, then fast, then slow again).
- If the *camera rotation* is activated, the viewport camera is rotated around the origin of the viewport along the y-axis. This feature enhances already animated exploded views and provides a simple animation of static ones by adding a 3D context.

## 5. Results

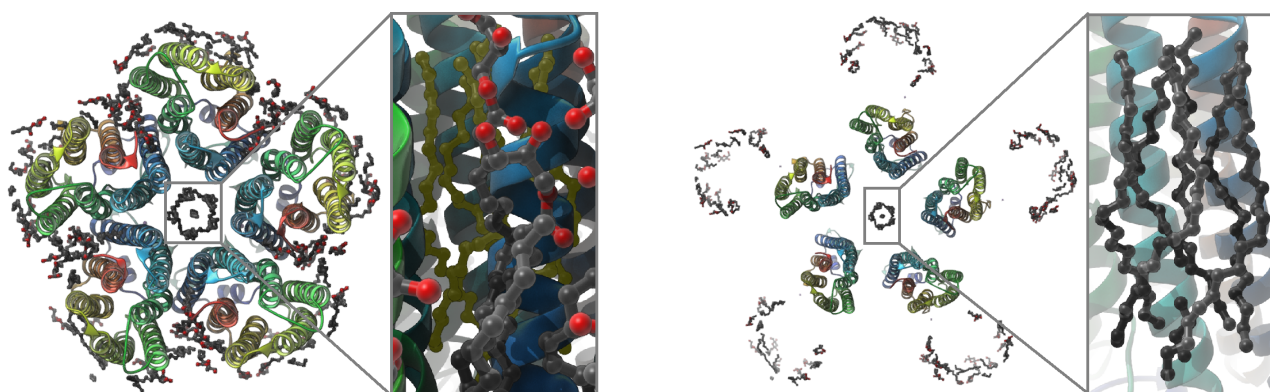
In this section, we present three use cases of our approach. We demonstrate the features introduced in Section 3 and 4 with two molecular structures and two DNA nano-structures, both of them being DNA origami structures. The use cases are based on the requirements of a domain scientist who is specialized in the development of molecular structures. The use cases focus on enhancements in the design process of molecular structures and how exploded views can be used to present such structures. The proposed approach is regarded valuable for the inspection of complex molecular structures and to model molecular processes.

### 5.1. Exposure of Occluded Components

During the design process of complex DNA nano-structures, occlusion of components that are currently worked on is a major problem for domain scientists. Using exploded views, occluding parts can be



**Figure 6:** Eigenvector exploded views of a DNA origami robot along the second eigenvector that isolates its scaffold from the staples.



**Figure 7:** Spherical exploded view of the crystal structure of the light-driven sodium pump KR2 protein (PDB 6RF3) [KPG<sup>+</sup>19]. In the original state (a) the protein, visualized using the secondary structure visualization provided by SAMSON [ND16], and the ligand (grey and red atoms) components are not well distinguishable. The exploded view (b) illustrates the single components distinctively.

moved easily, which creates an improved working environment for the designer. We demonstrate this on the DNA origami robot used in the work of Castro et al. [CKK<sup>+</sup>11] and shown in Figure 6a.

The robot is a rather complex DNA origami design. In the example, we show how the staples can be removed to reveal the scaffold. Since the robot has more of a linear than a spherical shape, we create an eigenvector explosion, since it is better suited for objects of this shape than a spherical explosion. As can be deduced by the orientation of the robot, its first eigenvector points along its vertical axis, the second one along the horizontal axis and the third one along the depth axis. We chose the second eigenvector, since it makes the most sense for our goal to unveil the scaffold. We additionally grouped the staples into five peeling groups. As seen in Figure 6b, the first layer of staples has been removed, by adjusting the explosion slider. If the domain expert who is working on this structure is currently only interested in the head of the robot, this view provides him or her with the exposed scaffold of the head. By moving the explosion slider to its maximum, we get a fully ex-

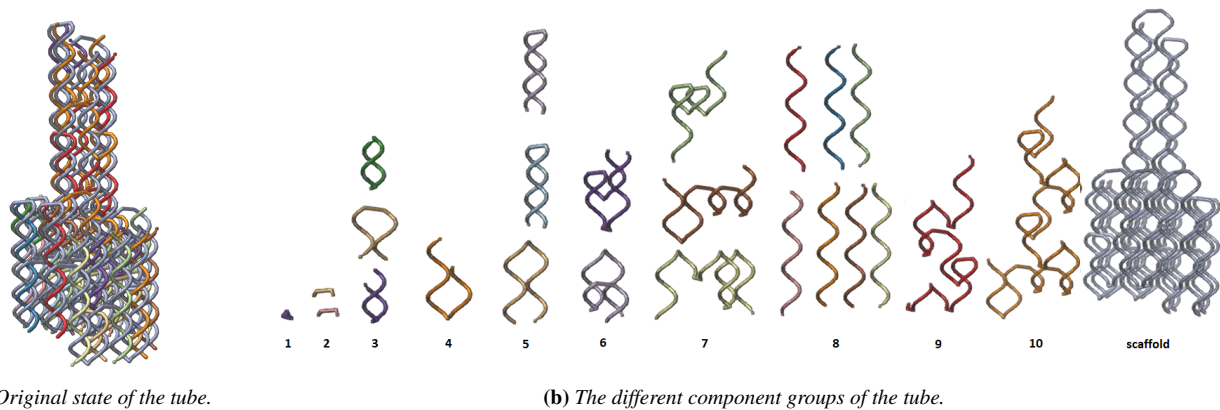
ploded view of the robot showing its scaffold in the center and the staples on its left and right (Figure 6c).

Generating exploded views as described in this example, allows domain experts to expose certain parts of a structure, while still maintaining information about components that occluded the part of interest before. An alternative approach to isolate the part of interest would be to displace only this part and therefore remove it from the occluding environment.

## 5.2. Inspection of Structure Composition

Due to the complex component-wise entanglement of molecular structures and DNA nano-structures, the inspection of their composition as well as relationships between single components is often a difficult task. We demonstrate how exploded views can help domain scientists in those tasks with the aid of two examples.

First, we take a look at a sodium pump KR2 protein [KPG<sup>+</sup>19]. To get a better insight into the pump's structure, we are interested



(a) Original state of the tube. (b) The different component groups of the tube.  
**Figure 8:** Free-form exploded view, that displays all groups of structure components of a DNA origami tube. The groups are sorted by their size from small (left) to big (right).

in the number and positions of its components and especially in the ligand in the center of the pump. In its original state (Figure 7a) the desired information is not readily visible. By generating a spherical explosion of the pump, we now can identify its main components and where they are bound together in the original state (Figure 7b). Since the occlusion for each component is also drastically reduced, we now can inspect the ligand in the center more deeply.

In the second example, we want to take a closer look at the components that comprise a DNA origami tube. As seen in Figure 8a, the tube consists of a scaffold and several staple strands. To get an understanding of the different staple types that are used in this tube, we create a free-form explosion that shows each group aligned horizontally. By first creating a spherical exploded view, we identify ten groups of staples and the scaffold. By drawing arrows to the desired locations in the viewport, we placed the staples of the individual groups above each other and aligned the groups horizontally. Finally, we hid the arrows to create a clear and uncluttered view. As seen in Figure 8b, for some groups, like group 3, 5 and 7, it is not immediately obvious that the staples of these groups have the same shape. This is because our approach does not provide the opportunity to rotate objects. Therefore, we cannot display the staples in a way that makes their similar shapes obvious. Nevertheless, this view provides a meaningful summary of the DNA origami tube's components and allows domain scientists to present its composition to others by showing us its components in sorted groups.

### 5.3. Simulation of Chemical Processes

A good way to depict chemical processes is through animation. Using our approach, simple processes can be visualized by first creating exploded views of the resulting structures and then animating the implosion of the view. We illustrate the Miller-Urey experiment [Mil53] using the proposed spherical explosion.

The goal of this experiment is to get a better understanding of the chemical origins of life. To achieve this, Miller and Urey created an environment that resembled the one on Earth before life was

present using only water ( $H_2O$ ), methane ( $CH_4$ ), ammonia ( $NH_3$ ), and hydrogen ( $H_2$ ). These molecules were put into a container half full of liquid water and then heated. After a while, amino acids began to form, showing that it is possible to create complex organic structures from simple inorganic ones. In the original experiment, Miller and Urey were able to produce five amino acids.

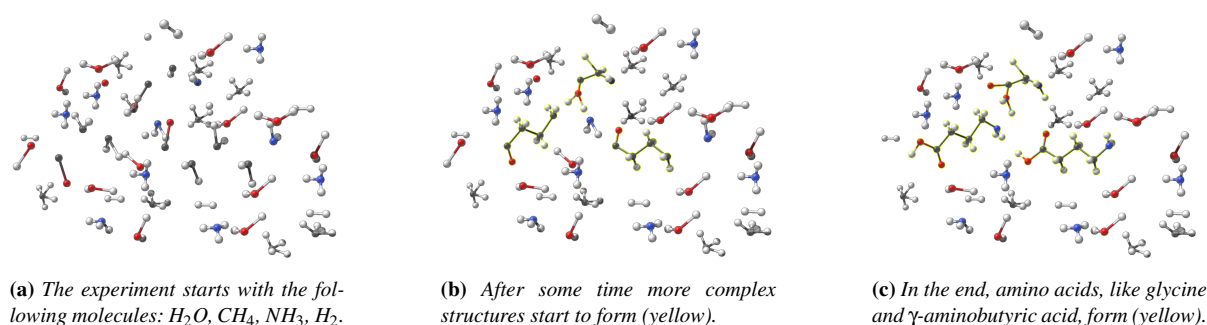
To simulate this experiment, we first design two of the amino acids that resulted: glycine and  $\gamma$ -aminobutyric acid. Next, we place several instances of the four involved molecules around the amino acids to simulate the used environment. We then explode all the atoms of the amino acids sequentially using *spherical explosion*. The resulting view is given in Figure 9a. This is the starting point of our simulation. By animating the reverse explosion of the current view or by manually adjusting the explosion slider, the user is now able to simulate how the amino acids that we created before form from a cloud of molecules. In Figure 9b, an intermediate stage of the reverse-explosion is shown. In this view, one can clearly see that complex structures start to form. Figure 9c illustrates the end product of the experiment with the amino acids in their final state.

By using spherical explosion on the amino acid atoms, the animation displaces the atoms linearly. Although the resulting trajectories do not depict how the atoms would move in vivo, we can create meaningful simulations to present simple chemical processes. For more complex simulations, a free-form exploded view is suggested.

### 6. Implementation and Performance

Our approach was implemented as a SAMSON element [ND16]. SAMSON is a well-established platform used for the prototyping process in computational nanoscience. The proposed method is implemented in C++. We use SAMSON's capability to load and interact with molecular data to implement our approach. For displaying the explosion paths as arrow sequences, SAMSON's capability for generating geometric primitives is used. The dataset hierarchy we employ in the structure-derived exploded view (Section 3.1) can be





**Figure 9:** Simulation of the Miller-Urey experiment.

extracted from the SAMSON data graph. To calculate the eigenvectors for the eigenvector explosion (Section 3.1.5), the library eig3 by Barnes [Bar07] is used. The user interface comprises a view of all the interactions provided for generating exploded views that were mentioned in Sections 3 and 4. Our system is implemented on a workstation with an AMD Ryzen 7 2700X 3.7GHz CPU, 32GB RAM and NVIDIA GeForce GTX 1080 graphics processor.

## 7. Conclusion and Future Work

In this paper, we provide a novel approach for handling visual clutter and occlusion if visualizing molecular structures. As stated by Kozlíková et al. [KKF<sup>+</sup>17], exploded views for handling visual clutter in biomolecular visualizations have not been widely used in state-of-the-art approaches. To the best of our knowledge, only Sorger et al. [SMR<sup>+</sup>17] and Furmanová et al. [FBG<sup>+</sup>18] also use exploded views when tackling these problems. We present two different strategies i.e., *structure-derived* and *free-form* exploded views. The main issue here is to derive the four key parameters of an exploded view: explosion distance, direction, order, and the selection of explosion components (see Table 1).

For the structure-derived exploded views, we derive these parameters directly from the spatial structure of the object at hand. The selection of the explosion components is handled through the existing hierarchy levels of the object. The explosion distance is calculated relative to the object's size and the actual distance can be chosen by the user via a slider. We also introduced three possible explosion orders: parallel, peeling, and sequential. For the sub-style *spherical explosion*, the components are moved along a vector from the object's center to the component center. For the *eigenvector explosion* the components are moved along one of the object's three eigenvectors. The free-form exploded view allows us to specify these parameters interactively. Explosion components can be defined by selecting random groups of atoms or structures at a higher level and processing them as a single component. To define the explosion distance and direction, the users can draw an arbitrary explosion path into the viewport, which allows them to create complex exploded views (Figure 8b).

The main *interaction tool* for the user is the explosion slider, which lets the user manipulate the explosion distance directly. By automating this slider, we provide the possibility to animate the explosion or implosion. Incorporating ease functions, camera rota-

tion, and controls of the animation speed, we enhance the animations and provide an animation tool-set.

The current limitations of our work are mostly in the interaction domain. It is computationally very expensive to change the position of atoms in SAMSON. The achieved frame-rates thus vary between 5 and 20 frames per second depending on the size of the dataset and the number of atoms. This is a problem for larger datasets. The interactivity of the free-form explosion could be enhanced by adding splines as a possibility to create the explosion path. Methodology-wise, it would be interesting to incorporate some kind of blocking constraints to reduce collisions of components during an animation. One possible approach for incorporating such constraints could be to prohibit path crossings for the free-form exploded view. Another solution could be to prioritize the movement of one component, if two components collide or are entangled.

We discussed our approach with a domain expert and tested it on realistic scenarios as described in Section 5. This shows its applicability in creating a visual representation of molecular structures and DNA nano-structures for the inspection of structure component relationships. Through isolating single components, the design process, especially of DNA nano-structures, is improved.

## Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 686647 ("MARA") and MSCA No. 747985. This publication is further supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-2019-CPF-4108 and BAS/1/1680-01-01. This project has also received funding from the ILLVISA-TION grant by WWTF (VRG11-010). This paper was partly written in collaboration with the VRVis Competence Center. VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET – Competence Centers for Excellent Technologies (854174), which is managed by FFG.

## References

- [APH<sup>+</sup>03] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. Designing effective step-by-step assembly instructions. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 828–837. ACM, 2003. 3

- [Bar07] Connelly Barnes. Eigenvectors of 3x3 symmetric matrix. <http://barnesc.blogspot.com/2007/02/eigenvectors-of-3x3-symmetric-matrix.html>, Feb 2007. Visited Mai 2019. 9
- [BF08] Michael Burns and Adam Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 154:1–154:7, New York, NY, USA, 2008. ACM. 2
- [BG06] Stefan Bruckner and M Eduard Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 2006. 3
- [BH04] Ryan Bane and Tobias Hollerer. Interactive tools for virtual x-ray vision in mobile augmented reality. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 231–239. IEEE Computer Society, 2004. 2
- [CKK<sup>+</sup>11] Carlos Ernesto Castro, Fabian Kilchherr, Do-Nyung Kim, Enrique Lin Shiao, Tobias Wauer, Philipp Wortmann, Mark Bathe, and Hendrik Dietz. A primer to scaffolded dna origami. *Nature methods*, 8(3):221, 2011. 7
- [DBC12] Shawn M Douglas, Ido Bachelet, and George M Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*, 335(6070):831–834, 2012. 1
- [DFW87] Thomas De Fazio and Daniel Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, 3(6):640–658, 1987. 3
- [DKDA11] Nadezhda T Doncheva, Karsten Klein, Francisco S Domingues, and Mario Albrecht. Analyzing and visualizing residue networks of protein structures. *Trends in biochemical sciences*, 36(4):179–182, 2011. 3
- [Elm05] Niklas Elmqvist. Balloonprobe: Reducing occlusion in 3d using interactive space distortion. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 134–137. ACM, 2005. 3, 5
- [ET07] Niklas Elmqvist and Mihail Eduard Tudoreanu. Occlusion management in immersive and desktop 3d virtual environments: Theory and evaluation. *IJVR*, 6(2):21–32, 2007. 2
- [ET08] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, 2008. 2
- [FBG<sup>+</sup>18] Katarína Furmanová, Jan Byška, Eduard M Gröller, Ivan Viola, Jan J Paleček, and Barbora Kozlíková. Cozoid: contact zone identifier for visual analysis of protein-protein interactions. *BMC bioinformatics*, 19(1):125, 2018. 9
- [KBE09] Michael Krone, Katrin Bidmon, and Thomas Ertl. Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1391–1398, 2009. 3
- [KKF<sup>+</sup>17] Barbora Kozlíková, Michael Krone, Martin Falk, Norbert Lindow, Marc Baaden, Daniel Baum, Ivan Viola, Julius Parulek, and Hans-Christian Hege. Visualization of biomolecular structures: State of the art revisited. In *Computer Graphics Forum*, volume 36, pages 178–204. Wiley Online Library, 2017. 1, 3, 9
- [KLW89] Ehud Kroll, Ehud Lenz, and John R Wolberg. Rule-based generation of exploded-views and assembly sequences. *AI EDAM*, 3(3):143–155, 1989. 3
- [KPG<sup>+</sup>19] Kirill Kovalev, Vitaly Polovinkin, Ivan Gushchin, Alexey Alekseev, Vitaly Shevchenko, Valentin Borshchevskiy, Roman Astashkin, Taras Balandin, Dmitry Bratanov, Svetlana Vaganova, et al. Structure and mechanisms of sodium-pumping kr2 rhodopsin. *Science advances*, 5(4):eaav2671, 2019. 6, 7
- [LACS08] Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin. Automated generation of interactive 3d exploded view diagrams. In *ACM Transactions on Graphics (TOG)*, volume 27, page 101. ACM, 2008. 3
- [LAS04] Wilmot Li, Maneesh Agrawala, and David Salesin. Interactive image-based exploded view diagrams. In *Proceedings of Graphics Interface 2004*, pages 203–212. Canadian Human-Computer Communications Society, 2004. 3
- [LCD06] Thomas Luft, Carsten Colditz, and Oliver Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph.*, 25(3):1206–1213, July 2006. 3
- [LVRH07] Ove Daae Lampe, Ivan Viola, Nathalie Reuter, and Helwig Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1616–1623, 2007. 3
- [MDLS<sup>+</sup>18] Haichao Miao, Elisa De Llano, Johannes Sorger, Yasaman Ahmadi, Tadija Kekic, Tobias Isenberg, M Eduard Gröller, Ivan Barišić, and Ivan Viola. Multiscale visualization and scale-adaptive modification of dna nanostructures. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1014–1024, 2018. 3, 5
- [Mil53] Stanley L. Miller. A production of amino acids under possible primitive earth conditions. *Science*, 117(3046):528–529, 1953. 8
- [MK93] Riaz Mohammad and Ehud Kroll. Automatic generation of exploded view by graph transformation. In *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*, pages 368–374. IEEE, 1993. 3
- [MTB03] Michael J McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *Visualization, 2003. VIS 2003. IEEE*, pages 401–408. IEEE, 2003. 2
- [NC16] S. Douglas N. Conway. cadnano. <http://cadnano.org/>, 2016. Visited Mai 2019. 2
- [ND16] NANO-D. Samson – software for adaptive modeling and simulation of nanosystems. <https://www.samson-connect.net/>, 2016. Visited Mai 2019. 2, 6, 7, 8
- [RKSZ94] Thomas Rist, Antonio Krüger, Georg Schneider, and Detlev Zimmermann. Awi: a workbench for semi-automated illustration design. In *Proceedings of the workshop on Advanced visual interfaces*, pages 59–68. ACM, 1994. 3
- [Rot06] Paul WK Rothmund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297, 2006. 2
- [SCS04] Henry Sonnet, Sheelagh Carpendale, and Thomas Strothotte. Integrating expanding annotations with a 3d explosion probe. In *Proceedings of the working conference on Advanced visual interfaces*, pages 63–70. ACM, 2004. 3
- [SK83] Nadrian C Seeman and Neville R Kallenbach. Design of immobile nucleic acid junctions. *Biophysical journal*, 44(2):201, 1983. 2
- [SMR<sup>+</sup>17] Johannes Sorger, Peter Mindek, Peter Rautek, Eduard Gröller, Graham Johnson, and Ivan Viola. Metamorphers: Storytelling templates for illustrative animated transitions in molecular visualization. In *Proceedings of the 33rd Spring Conference on Computer Graphics, SCCG '17*, pages 2:1–2:10, New York, NY, USA, 2017. ACM. 3, 9
- [TCM06] Marco Tarini, Paolo Cignoni, and Claudio Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006. 3
- [TKS10] Markus Tatzgern, Denis Kalkofen, and Dieter Schmalstieg. Compact explosion diagrams. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 17–26. ACM, 2010. 3
- [VFSL02] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Antoni Llobet. Viewpoint entropy: a new tool for obtaining good views of molecules. In *ACM International Conference Proceeding Series*, volume 22, pages 183–188, 2002. 3
- [WC<sup>+</sup>53] James D Watson, Francis HC Crick, et al. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953. 2
- [ZNL14] Fei Zhang, Jeanette Nangreave, Yan Liu, and Hao Yan. Structural dna nanotechnology: state of the art and future perspective. *Journal of the American Chemical Society*, 136(32):11198–11211, 2014. 1, 2